

Extending JXTA Protocols for P2P File Sharing Systems

Fatos Xhafa

Polytechnic University of Catalonia
Department of Languages and Informatics Systems
C/Jordi Girona 1-3, 08034 Barcelona, Spain
Email: fatos@lsi.upc.edu

Leonard Barolli

Department of Information and Communication Engineering
Fukuoka Institute of Technology (FIT)
3-30-1 Wajiro-higashi Higashi-ku, Fukuoka 811-0295, Japan
Email: barolli@fit.ac.jp

Raul Fernández, Thanasis Daradoumis and Santi Caballé

Open University of Catalonia
Department of Information Sciences
Barcelona, Spain.
Email: {rfernandezco,adaradoumis,scaballe}@uoc.edu

Vladi Kolici

Department of Electronics
Polytechnic University of Tirana
Mother Teresa Square, Nr.4, Tirana, Albania

Abstract—File sharing is among the most important features of the today's Internet-based applications. Most of such applications are server-based approaches inheriting thus the disadvantages of centralized systems. Advances in P2P systems are allowing to share huge quantities of data and files in a distributed way. In this paper, we present extensions of JXTA protocols to support file sharing in P2P systems with the aim to overcome limitations of server-mediated approaches. Our proposal is validated in practice by deploying a P2P file sharing system in a real P2P network. The empirical study revealed the benefits and drawbacks of using JXTA protocol for P2P file sharing systems.

I. INTRODUCTION

Peer-to-Peer (P2P) systems have become popular due to file sharing among millions of user world wide. Since the appearance of Napster, Freenet and Gnutella, file sharing software has been a hot research topic for industrial and academic purposes. As a matter of fact, there are still many of issues to understand and better address in P2P file sharing domain. Thus, deciding which protocols to use [6] (Freenet, Gnutella, and Napster use different protocols), how to reduce the traffic generated due to file sharing [3], [4], what schemes to use for efficient indexing/searching files within a P2P file systems [2] are such research questions, to name a few. On the other hand, there are issues related to the design of P2P systems for file sharing that benefit from the decentralized nature of P2P systems yet facilitate efficient file sharing among peers and P2P file sharing applications. Indeed, there is a whole range from server-mediated architecture P2P systems to fully decentralized systems and appropriate architectures that match different needs of P2P file sharing applications are to be investigated.

Actually the most popular alternatives for file sharing are Kazza, Overnet and Bittorrent, each one with its own net and form of searching and indexing files. Kazza uses a big server to provide searching and indexing of the files. Overnet indexes the files in a list of different servers; the user is connected to one of them, through which are done all the searches. In the

Bittorrent net, the user has a little file with the specification of each download. There is a central node that organizes all the clients.

Unfortunately, all this alternatives are for a general sharing system and a user can't share a file or a folder with the grant that only a particular group of people can have it. One could use as an alternative sharing files with people that a user chooses through IM programs (like MSN Messenger), but in this case file sharing is automatic and doesn't permit that the destination user chooses what files wants from other users (download under demand). Thus, such file sharing systems are not appropriate for academic online campuses, where students working in small groups would like to share notes, do homework in group in real time or help each other. Moreover, these file sharing programs use a centralized architecture, needing either a server or a very special node to control all the net. This work is motivated to propose

Peer-to-Peer (P2P) systems are decentralized, self-organizing distributed systems showing each time more remarkable improvements in scalability, robustness and distributed storage. P2P networks are the alternative to the traditional client-server applications by replacing client-server communication with peer interactions; peers can serve as client, servers, edge peers providing thus much more flexibility; a peer can request files from others and share files with other peers. In the domain of P2P protocols, JXTA technology [1], [5], [7] is an interesting alternative in the file sharing field given that its protocols allow to develop P2P platforms, either pure or mixed. This property is certainly important since pure P2P need not the presence of a server for managing the net. Unfortunately, JXTA protocols do not support file sharing, however it offers a set of basic protocols, such as publication of advertisements, that can be further developed and used for file sharing purposes.

The objective of this work is to extend and improve the JXTA protocols to support the development and deployment

of P2P file sharing systems. Moreover, we would like such protocols to be practically useful also for academic context giving thus support to online learning teams. As mentioned above, there is a whole range of architectures from server-mediated to pure P2P networks. In the former architecture the central server is responsible for maintaining/indexing shared files and respond to requests for files while peer nodes are responsible to host the files and make available the information on shared files to the server. In the later architecture, there is no central server but peers that can play both client and server roles for sharing and downloading files. This work is also motivated by the investigation of a new P2P architecture, namely a broker-based P2P system and its benefits for P2P file sharing systems. The idea of P2P architectures using broker peers and client peers has already been exploited for task execution in P2P distributed systems [10].

Our proposal is validated in practice by deploying a P2P file sharing system in a real P2P network. The empirical study revealed the benefits and drawbacks of using JXTA protocol for P2P file sharing systems.

II. JXTA AS A BASIS FOR FILE SHARING SYSTEMS

JXTA is a generalized group of protocols that allow different devices to communicate and collaborate among them. JXTA offers a platform covering basic needs in developing P2P networks.

A. JXTA Protocols

For the communication between peers is used a group of asynchronous protocols based in the model request/reply. The different protocols standardizes the way to find other peers, the communication between them, the formation of groups, the publication and the discovery of different advertisements. As part of these protocols, there are functionalities called services (see Fig. 1): (a) *Rendezvous Service*: used for publishing messages outside the peerGroup; (b) *Discovery Service*: used to discover Advertisements (peers, peerGroups, pipes and other services) of a peerGroup; (c) *Pipe Service*: offers operations to send and receive data over the pipes; (d) *Endpoint Service*: to transmit simple messages (used in the old version 1.0); (e) *Resolver Service*: offers a generic mechanism to send requests and receive replies; (f) *Peer Info Service*: allows to obtain information about nodes of the group; and, (g) *Membership Service*: allows a peer to establish a unique identity in order to guarantee the presence in a group.

The above-mentioned services are the basis for the development of P2P file sharing systems. Moreover, JXTA offers other protocols for peer discovery, peer communication, publishing of advertisements and more generally discovering the information of other peers. Our starting P2P platform is the JXTA-based P2P platform [11], briefly presented next.

B. The JXTA-Overlay

JXTA-Overlay project is an effort to use JXTA technology for building an overlay on top of JXTA offering a set of basic primitives (functionalities) that are most commonly needed

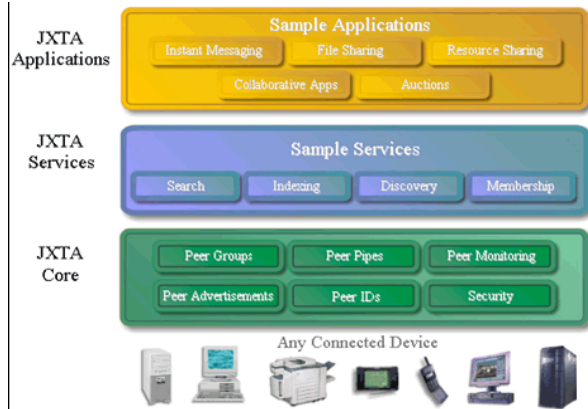


Fig. 1. The JXTA services.

in JXTA-based applications. The proposed overlay comprises primitives for:

- Peer discovery
- Peer's resources discovery
- Resource allocation
- Task submission and execution
- File/data sharing, discovery and transmission
- Instant communication " Peer group functionalities (rooms etc.)
- Monitoring of peers, groups, tasks etc.

This set of basic functionalities is intended to be as complete as possible to satisfy the needs of JXTA-based applications. The overlay is built on top of JXTA layer and provides a set of primitives that can be used later by other applications, which on their hand, will be built on top of the overlay, with complete independence. The JXTA-Overlay project has been developed using the latest updated JXTA libraries. In fact, the project offers also several improvements of the original JXTA protocols/services to increase the reliability of JXTA-based distributed applications.

III. EXTENSION OF JXTA PROTOCOLS FOR FILE SHARING

The JXTA-Overlay already offers several improvements of JXTA protocols, including:

- *Presence management*: Periodically the PeerAdvertisement is published and the local cache is synchronized.
- *Connection group*: a peer is connected to the general groups through a broker. The different brokers available are known by a previously published list. Furthermore, the user is connected automatically to the different groups he belongs to.
- *Communication mechanism*: JXTA-Overlay provides an error control in message sending of and organizes the net in such a way that every peer can send a message in a straightforward to any peer by only knowing the peer name.
- *Incoming messages mechanism*: the messages are queued in peer's incoming queue in order to ensure messages are

received and correctly treated at the peer's side.

In spite of the improvements provided by JXTA-Overlay, there are still two limitations for using it as a basis for P2P file sharing systems. The first limitation has to do with the *treatment of the advertisements*.

a) *The treatment of the advertisements*: The original JXTA-Overlay processes all the advertisements received by a peer just at the time they are received. If we would use the advertisements for publishing information on files shared by a peer, this could be very problematic. Indeed, the number of files shared by a peer could be large and therefore the computational cost would be very high (prohibitive for practical purposes). Moreover, as the file sharing advertisement would be processed at the at the time they are received by a peer, this could cause peer collapse.

The solution to this problem is to treat the advertisements only whenever their information is needed. When an advertisement is received, its information is saved in the original XML format if it is really needed, and then it is treated. This treatment is done only for the first time, because when it is treated, the information is saved as it is done in the original JXTA-Overlay. This idea is also applicable to the opposite direction: when an advertisement has to be published, if the XML is already built it is published; if not, the XML is built only the first time (see Fig. 2).

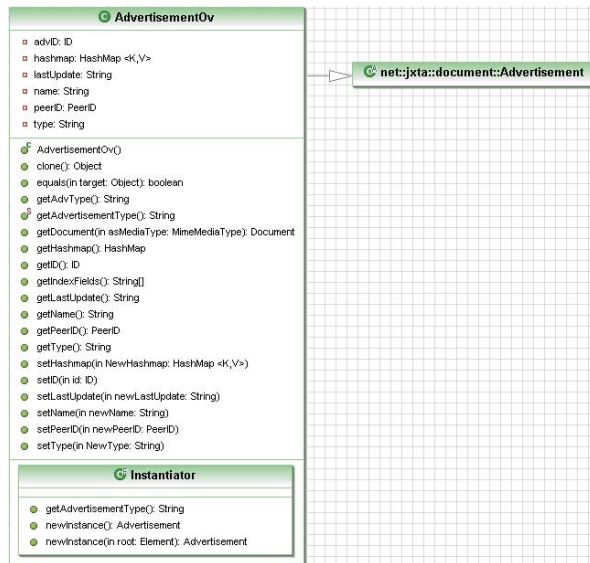


Fig. 2. Diagram of the Advertisement's Data Container.

b) *Publishing of the advertisements*: In the JXTA-Overlay, the file advertisements are send very frequently (within very short time intervals) and have a short lifetime. This is done so in order to ensure that only the online peers will have advertisements operative. However, if a peer shares a large quantity of files, the file advertisement will be large and computationally costly to publish and to treat; again, publishing the file advertisement very frequently could certainly be

problematic. Our solution is that the file advertisements should be periodically send in rather long time intervals solving thus the large quantity of files to be processed. Yet, with this specification, advertisements must have a longer lifetime implying that if a peer disconnects, its advertisements continue in the net. In order to deal with this problem, we use broker peers: whenever a broker works with the file advertisements, it always checks that the user is connected, checking thus the existence of the information advertisement (and its main attributes) that every peer sends to the net to communicate that it is online.

IV. IMPLEMENTATION OF THE P2P FILE SHARING SYSTEM

Once the JXTA protocols were re-implemented, they were used to implement a file sharing system as part of the JXTA-Overlay (see Fig. 3).

A user can share different files with its peerGroup. When the user shares a file, the overlay computes its entire *attribute* and the configuration is saved in the file `./userData/username/clientConf/groupname`. The next time that the peer joins the overlay, this configuration will be loaded and the files will be checked for possible changes. Every pre-fixed interval of time (15 minutes by default) the information on shared files are collected advertisements to be published in the net.

Group / File	Folder	Size	Type	Description type	Last modification
Total	8 files	14.9 MB			
net:jxta:document:Advertisement	8 files	14.9 MB			
net:jxta:document:Advertisement	C:\Documents and Settings\user\My Documents	6.1 MB	audio	mp3	1.3559272140
net:jxta:document:Advertisement	C:\Documents and Settings\user\My Documents	6.2 MB	application	pdf	1.3559272140
net:jxta:document:Advertisement	C:\Documents and Settings\user\My Documents	70 KB	text	txt	1.3559272140
net:jxta:document:Advertisement	C:\Documents and Settings\user\My Documents	8 KB	image	png	1.3559272140
net:jxta:document:Advertisement	C:\Documents and Settings\user\My Documents	204 KB	image	gif	1.3559272140
net:jxta:document:Advertisement	C:\Documents and Settings\user\My Documents	204 KB	text	rtxt	1.3559272140
net:jxta:document:Advertisement	C:\Documents and Settings\user\My Documents	1.9 MB	audio	mp3	1.3559272140
net:jxta:document:Advertisement	C:\Documents and Settings\user\My Documents	204 KB	text	rtxt	1.3559272140

Fig. 3. Snapshot of JXTA-Overlay file sharing.

c) *File searching*: A user can search a file with a few specified parameters with the objective to find certain files that are shared in the net. When the user introduces the parameters, the peer client sends this request to the broker of the peerGroup. Upon receiving the request, the broker will find in the correct advertisements all the files that the users share, according to the search parameters. Every advertisement has the files that a user shares in a peerGroup. As we have already mentioned (see Sect. III, Paragraph III-0b), the broker needs to assure that the peer is connected. To this end, the broker checks if there is a "Client Peer Information Advertisement" of this peer and in case the broker finds it, it means the peer is connected since this sort of advertisements have a very short lifetime. Once the broker checked the advertisement of a peer, it will start checking one by one all the files that this peer shared in this peerGroup and will save in a list all the files that have all the attributes that the user have requested. Once the broker had checked all the files of all the groups, it sends the complete list of the found files to the peer that did the original request, which shows this information to the user in a table, from which the user can request to download different files.

V. EXPERIMENTAL STUDY

A. Deployment of the P2P JXTA-Overlay

In order to conduct a first experimental evaluation we deployed our P2P file sharing system in a cluster of machines at our university campus. We show in Table I the specification of the machines that we used to deploy the net.

TABLE I
PEER NODE SPECIFICATIONS

Peer node	Characteristics
Broker	Double dual core (4 CPUs with HT (8 CPUs))
Client 1	Double dual core (4 CPUs with HT (8 CPUs))
Client 2	Double dual core (4 CPUs with HT (8 CPUs))
Client 3	Double dual core (4 CPUs with HT (8 CPUs))
Client 4	Double dual core (4 CPUs with HT (8 CPUs))
Client 5	Double dual core (4 CPUs with HT (8 CPUs))
Connection	1 Gbps

B. Experimental scenarios

The objective of the experiment is to empirically study the performance of the P2P file sharing system regarding a few parameters of the file search. In fact, we implemented to search engines: brute force, that is, searching files in peer nodes without any file classifications and, file search by group, in which files are grouped by the P2P system according to their type (Txt, Doc, Mp3, ...); these two types of searches are called search of type A and type B, respectively (see also Table II). We studied, among others, the following parameters:

- Advertisement publishing time
- Broker processing time of advertisements
- Time efficiency of the file search according to different search criteria
- Bytes (size) of the advertisements
- Number of files found
- CPU and memory consumption

TABLE II
FILE TYPES USED IN THE EXPERIMENTAL STUDY

Extension	Description	Group
Txt	Text file	Text
Doc	Microsoft Word file	
Mp3	MP3 file	Audio
Wma	Windows Microsoft Audio	
Avi	DIVX file	Video
Mpg	MPEG-2 file	
Jpg	Compressed image	Picture
Gif	Animation image	
Zip	Compressed file	Application
Rar	Compressed file	

We have measured these parameters in 10 different tests for each possible combination of the 3 components that can affect the search:

- 1) Number of files that share each peer: between 0 and 300; in groups of 50 files (10 for each possible type: audio, video, picture, application and text).
- 2) Number of Peers: between 1 and 5.

- 3) Type of search: brute force and file type search we have 2 different types of search: type A and type B (see above).

C. Computational results

We report here some preliminary computation results. The first parameter to study was the file advertisement in our JXTA-based P2P network. Indeed, the JXTA protocols were extended to support file sharing. However, it should be mentioned that the parameters related to file advertisements, that is, the time interval (periodicity) to publish the file advertisements and their lifetime are critical to the P2P file sharing system.

d) *Advertisement publishing time:* We show in Fig. 4 the relation between the publication time with the increasing number of shared files in the peer nodes and in Fig. 5 the relation between the advertisement size and the number of shared files. As can be seen from the figures, while the number of files increases the publication time of one advertisement seem to increase exponentially (Fig. 4) while its size grows linearly (Fig. 5) with the increasing number of shared files.

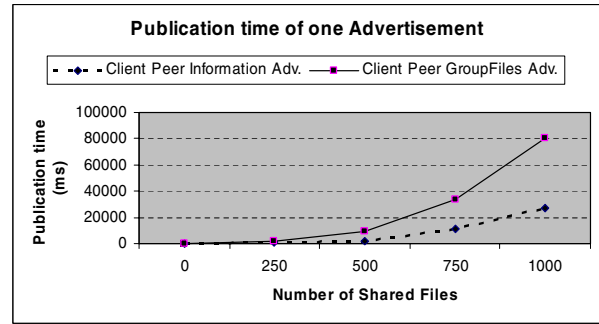


Fig. 4. Publication advertisement time.

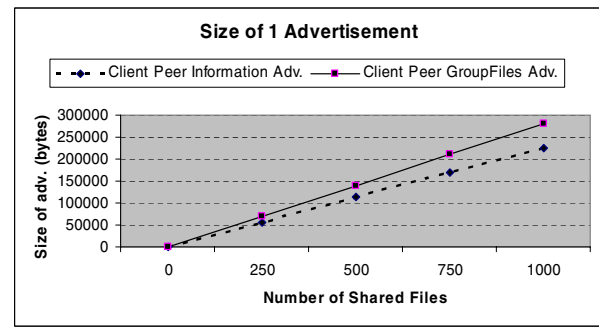


Fig. 5. Advertisement size.

Certainly, the exponential growth in the publishing time of advertisements is prohibitive if a large number of files are to be shared by peers. In order to alleviate the burden of the publishing time, we re-structured the information contained in an advertisement. More precisely, we divided the general information on a peer and from the information on its shared files,

which originally were together in the same advertisement, namely, the Client Peer Information Advertisement. So the advertisement are now separated in: *Client Peer Information Advertisement* containing the general peer information, *Client Peer Files Advertisement* containing information about all files shared by a peer and *Client Peer GroupFiles Advertisement* containing the information of shared files grouped according to different file types. The objective of this separation was twofold: on the one hand to reduce the publishing time of the advertisement and, on the other, to separate the *lifetime* of publishing advertisements on peer information from the lifetime of publishing the advertisements on the files shared by the peer. With this separation we can now tune the lifetime parameters so that we can publish the first advertisement every few seconds (experimentally fixed to 5 seconds) and the other two advertisements (on files) every few minutes (experimentally fixed to 15 minutes). We show the graphical representation of the separation of advertisement in Figs. 6 and 7.

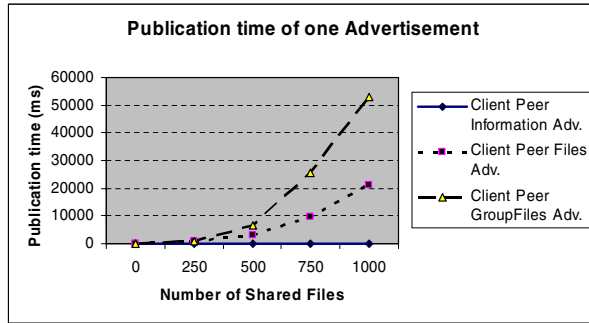


Fig. 6. Publication advertisement time.

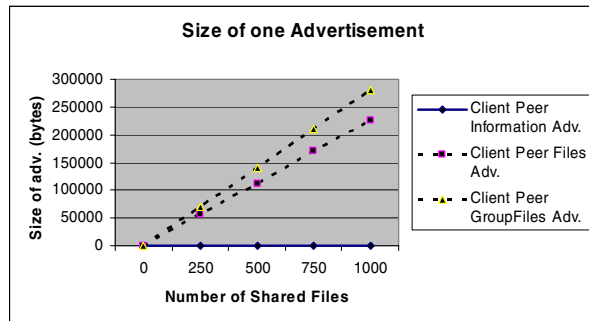


Fig. 7. Advertisement size.

It should be also noted that in the Client Peer Files Advertisement is published only the name, size and hash for each shared file while in the Client Peer GroupFiles Advertisement, each file has associated the group, a description of the group as well as its last modification. This implies that the Client Peer GroupFiles Advertisement publishes much more information per shared file and therefore requires more time for processing

and publishing in the P2P net.

We also studied the comparative advertisement publishing time of advertisement when just two advertisement were implemented (referred to as type A) vs. three advertisement (referred to as type B), as shown in Fig. 8.

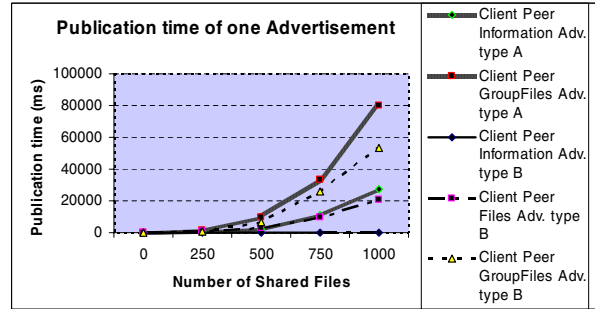


Fig. 8. Comparison of publication advertisement times.

From Fig. 8 we can appreciate some differences in the publishing time, that is, the publishing time of advertisement of type B is smaller than the publishing time of advertisement of type A, confirming that it was worth the separation of advertisements.

e) Broker processing time of advertisements: We recall that the architecture of the JXTA-Overlay platform is based on broker peers (the governors of the P2P net) and client peers (edge peers either with a graphical user interface or just cmd line). Broker peers play an important role since they are in charge to control the P2P net; in particular, broker peers process search peers' file advertisements as well as requests for file search. Therefore, it is important to study the processing time of file advertisement at broker side. Three parameters were considered for this study:

- The number of shared files by each peer client
- The number of peers clients
- The type of the search

We show in Fig. 9 the graphical representation of the processing time of advertisement by a broker peer as the number of shared files increases and different numbers of client peers are connected to the net and files of a specified type are searched. The similar graphical representation when five types of files are searched is shown in Fig. 10.

As can be seen from Figs. 9 and 10, the broker's processing time of file advertisements increases linearly when the number of peers in the net increases. On the other hand, we observe a threshold value on the processing time depending on the number of connected peers and the number of shared files. More precisely, when three peers are connected, the broker processing time of file advertisements increases linearly when the number of shared files ranges from 1 to 250 and after that value an important increment is observed (the threshold value is 250 files in this case); when four peers are connected, the broker processing time of file advertisements increases linearly for the number of shared files between 1 and 200 and after that

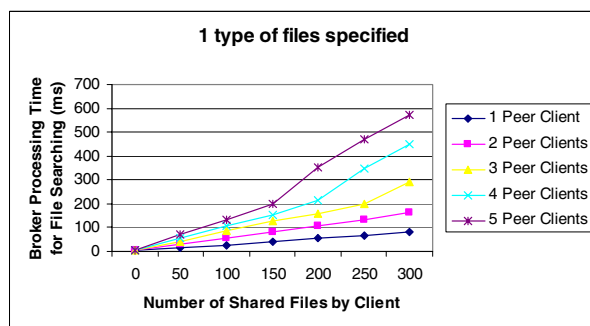


Fig. 9. Broker's processing of file advertisement (only files of a specified type are searched).

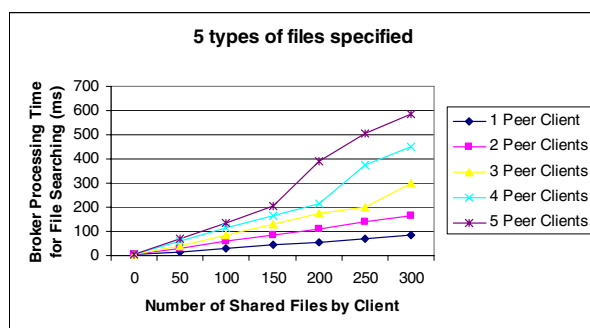


Fig. 10. Broker's processing of file advertisement (only files of two specified types are searched).

value an important increment is observed (the threshold value is 200 files in this case). As the number of peers increases, the threshold value decreases (it is equal to 150 for five connected peers), meaning that the broker processing time of file advertisement increases significantly.

VI. CONCLUSIONS

In this paper we have presented extensions of JXTA protocols to support development of file sharing systems in JXTA-based P2P applications. The extension of the basic protocols of JXTA required the definition and management of file advertisements to enable file sharing as well as efficient searching of files.

Several issues were encountered in extending the JXTA protocols for file advertisement. Among them, the efficient management of advertisement was identified and studied. Indeed, file advertisement could signify an important computational burden to broker peers due to the large size of file advertisements and the large quantity of file advertisements published by different peers that must be processed by the broker. Another important issue is that of the *lifetime* of file advertisements in order to increase the reliability of the file systems in the P2P net. Certainly, short values of lifetime would imply very frequent publishing of advertisements, which on turn, could provoke broker peers collapse; on the other hand, large values of the lifetime would imply the

“existence” of files in the P2P net while peers sharing the files are already disconnected from the net. Our approach proposes a separation of types of advertisement in peer proper information advertisement, file advertisement and group files advertisement. This separation allows for an adequate fine tuning of the lifetime parameter according to the type of advertisement. By this approach we are able to alleviate the computational load of broker as well as increase the reliability of the file systems in the P2P net.

The approach has been experimentally studied by deploying the JXTA-based P2P network in a real small network at the virtual campus of our university.

In our future work we plan to complete the experimental study by considering a larger P2P network. Also, we would like to study the differences between P2P file sharing and Web traffic of server-mediated file sharing approaches, which could reveal important differences in both approaches. In the same context, it would be interesting to study possible bandwidth savings in JXTA-based P2P file-sharing architectures. Finally, we are interested to apply our JXTA-based P2P file sharing system to support cooperative work of online teams at our virtual campus.

ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish MCYT project TSI2005-08225-C07-05.

REFERENCES

- [1] D. Brookshier, D. Govoni, N. Krishnan, and J.C Soto. *JXTA: Java P2P Programming*. Sams Publishing, 2002.
- [2] F. Le Fessant, S. Handurukande, A. M. Kermarrec, L. Massoulié. Clustering in Peer-to-Peer File Sharing Workloads *Peer-to-Peer Systems III*, Lecture Notes in Computer Science, Vol. 3279/2005, 217–226, 2005
- [3] K.P. Gummadi and R.J. Dunn and S. Saroiu and S. D. Gribble and H.M. Levy and J. Zahorjan. *Measurement, modeling, and analysis of a peer-to-peer file-sharing workload*. Proceedings of the nineteenth ACM symposium on Operating systems principles, 314–329, ACM Press, New York, NY, USA, 2003.
- [4] Th. Karagiannis, A. Broido, M. Faloutsos and Kc Claffy. Transport layer identification of P2P traffic. *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, 121–134, Taormina, Sicily, Italy, ACM Press, New York, NY, USA, 2004.
- [5] S. Li. *Early Adopter JXTA*. Wrox Press Information Inc., 2003.
- [6] S. Man Lui and S. Ho Kwok. *Interoperability of peer-to-peer file sharing protocols*. SIGecom Exch.Journal, **3**(3), 25–33, ACM Press, New York, NY, USA, 2002.
- [7] S. Oaks, B. Traversat, and L. Gong. *JXTA in a Nutshell*. O'Reilly, 2003.
- [8] J. A. Pouwelse, P. Garbacki, D. H. Epema, and H. J. Sips. The Bittorrent P2P File-sharing System: Measurements and Analysis *Proceedings of the 54th International Workshop on Peer-to-Peer Systems (IPTPS'05)*, Ithaca, USA, 2005.
- [9] J.E. Riasol and F. Xhafa. Juxta-cat: a jxta-based platform for distributed computing. *Proceedings of the 4th Int. Symposium on Principles and Practice of Programming in Java*, 72–81, 2006. ACM Press.
- [10] F. Xhafa, R. Fernandez, Th. Daradoumis, L. Barolli and Santi Caballé. Improvement of JXTA Protocols for Supporting Reliable Distributed Applications in P2P Systems. In *Proceedings of Network-Based Information Systems, First International Conference, NBIIS 2007, Regensburg, Germany, September 3-7, 2007*, Lecture Notes in Computer Science, Vol. 4658, 345–354, Springer, 2007.
- [11] F. Xhafa and R. Fernández. Jxta-overlay: <http://jxta-overlay.dev.java.net>